

RotoSwepe: Word-Gesture Typing using a Ring

Aakar Gupta, Cheng Ji, Hui-Shyong Yeo, Aaron Quigley, Daniel Vogel*
Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada
(aakar.gupta,c7ji,dvogel)@uwaterloo.ca
University of St. Andrews, St. Andrews, UK
(hsy,aquigley)@st-andrews.ac.uk

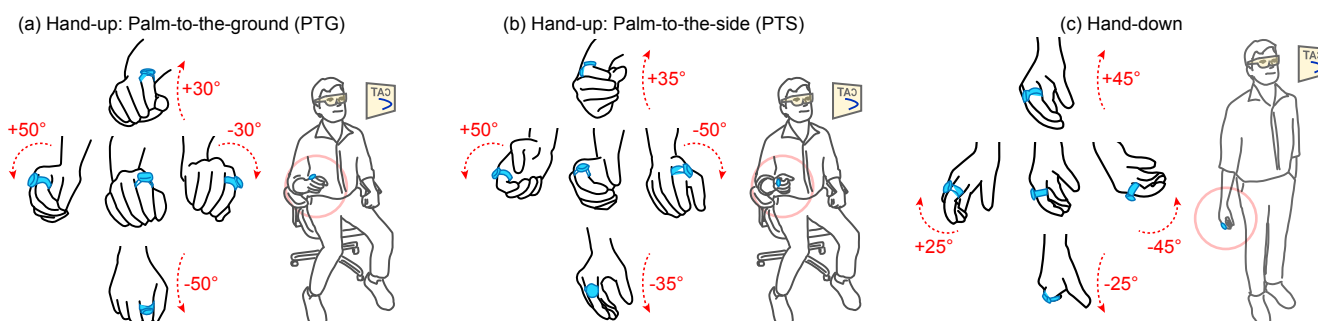


Figure 1: RotoSwepe ring-based word-gesture typing for AR: candidate hand postures with rotation ranges.

ABSTRACT

We propose RotoSwepe, a technique for word-gesture typing using the orientation of a ring worn on the index finger. RotoSwepe enables one-handed text-input without encumbering the hand with a device, a desirable quality in many scenarios, including virtual or augmented reality. The method is evaluated using two arm positions: with the hand raised up with the palm parallel to the ground; and with the hand resting at the side with the palm facing the body. A five-day study finds both hand positions achieved speeds of at least 14 words-per-minute (WPM) with uncorrected error rates near 1%, outperforming previous comparable techniques.

CCS CONCEPTS

• Human-centered computing → Interaction tech.

KEYWORDS

interaction techniques, controlled experiments

ACM Reference Format:

Aakar Gupta, Cheng Ji, Hui-Shyong Yeo, Aaron Quigley, Daniel Vogel. 2019. RotoSwepe: Word-Gesture Typing using a Ring. In *CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019)*, May 4–9, 2019, Glasgow, Scotland Uk. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3290605.3300244>

1 INTRODUCTION

Text-entry for head-mounted devices (HMDs) for augmented reality (AR), virtual reality (VR) remains an open problem. Existing approaches introduce solutions that use hand-held controllers [22, 63] or swiping on smartglasses [1, 25, 82]. However, hand-held controllers are not feasible for AR, especially in mobile scenarios. Swiping on smartglasses solves that problem, but current techniques report up to 10 WPM for expert use, which requires improvement. A second problem with these approaches is that the techniques are optimized for a dedicated device and do not allow for cross-device text-input. Controllers work best with VR, swiping on the edge of smartglasses is specific to touch-sensitive AR HMDs, and none of these techniques can work with other smart devices such as a distant screen or a smart-watch or an IOT (Internet-of-Things) device. An ideal technique would be one which can be useful across these device scenarios. While speech is an obvious solution, it can be error-prone for certain users and situations, and can be socially awkward.

We propose RotoSwepe, a technique that inputs text using a ring in mid-air. RotoSwepe uses ring-motion, and consequently wrist-motion, to draw a motion trace over the keyboard, which types a word. Shape-writing or word-gesture typing (WGT) was introduced by Kristensson and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org. *CHI 2019*, May 4–9, 2019, Glasgow, Scotland Uk

© 2019 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-5970-2/19/05...\$15.00

<https://doi.org/10.1145/3290605.3300244>

Zhai [40, 41, 83] wherein users write a word by tracing a path traversing the letters on the keyboard using a stylus or a finger. Instead of a finger, RotoSwipe uses the ring’s tilt-motion to trace a path over the keyboard. RotoSwipe is the first ring-based technique that enables word-gesture typing. RotoSwipe has the following characteristics: a) *Unencumbered*: RotoSwipe does not require a hand-held device, but a finger-worn ring with a simple motion sensor that allows a miniature design. However, it is not completely *freehand* [7], given the ring, and therefore we term its use as *unencumbered*. b) *One-handed*: While typing, the input is completely one-handed, thus allowing greater flexibility in on-the-go scenarios or when the second hand is encumbered. c) *Self-contained*: RotoSwipe is self-contained and does not need a surface to provide input, enabling the user to type in mid-air, in any posture, sitting or standing. d) *Eyes-away*: RotoSwipe does not need the user to pay attention to the text-input device, freeing them to look at the keyboard, wherever it may be. e) *Any-qwerty input*: Using word-gesture typing along with a qwerty keyboard means that RotoSwipe can work with any existing qwerty keyboard on any device including HMDs for AR/VR. Given the lock-in effect of qwerty today [8], it is essential for a text-entry technique to support qwerty-based typing.

In this work, we design and implement RotoSwipe to work with a standard Android keyboard. We support two hand postures: when the hand is above the waist; and when it is below the waist, relaxed. Since ring orientation changes completely for these two postures, we design two slightly different styles of text-entry for the two postures, which we refer to as - *Hand-Up* and *Hand-Down*. For *Hand-Up*, we further compare for performance between two different mappings of ring orientation to the keyboard plane. Finally, we conduct a five-day study for both *Hand-Up* and *Hand-Down* postures with a HMD. The results show that with <60 mins of typing, both postures result in a speed of >14 WPM, with a near 1% uncorrected error rate, outperforming current techniques that do not use a hand-held controller. The results further showed no signs of plateauing indicating the scope for further improvement.

We make the following contributions: 1) Propose and implement a ring-tilt based word-gesture typing technique that enables unencumbered, one-handed qwerty typing for HMDs (and other devices). 2) Demonstrate using a 5-day study that RotoSwipe is fast, accurate, and easy to learn.

2 BACKGROUND AND RELATED WORK

Tilt-based Text-entry for Handheld Devices

A subset of text-entry techniques proposed for handheld devices (feature phones, smartphones, controllers) can potentially be useful for AR/VR scenarios. While there are

eyes-free techniques that rely on touch gestures or tapping input on custom layouts [6, 9, 46], many potentially useful techniques incorporate tilt. Tilt-based text-entry has been explored with the following characteristics: discrete/continuous tilting, custom/standard (12-key, qwerty) layouts, and letter-entry/word prediction. Tilttype [57] uses multi-step discrete 8-directional tilting to enter letters on small screens. Tilttext [74] uses discrete 4-directional tilting to disambiguate the letter after keypress on the standard 12-key keypad. GestText [35] uses discrete 8-directional tilting with a Wiimote controller on a custom layout for letter-entry. Castellucci et al. [12] investigate H4Writer, a Huffman code-based layout which uses 4-directional Wiimote tilting for text entry. Unigesture [60] and Hex [75] use a sequence of discrete tilting gestures on a custom layout with word groupings and then infer the desired word. While WGT using touchscreens has been heavily investigated, WGT or similar approaches using continuous tilt to control a path over the letters have been rarer. RoText [71] maps a smartphone’s 1D tilt range to a 1D letter layout and predicts words based on the user’s tilt motion along the range. Technically, the closest work to RotoSwipe is Yeo et al’s SWiM [78], that uses 2D continuous tilt movements of the smartphone to generate a WGT trace on the smartphone’s qwerty keyboard. Participants achieved speeds of 15.5 WPM after 3 blocks. However, SWiM requires a hand-held device and hence encumbers the hand.

Mid-air text-entry without handheld devices

Techniques that support typing while the hand is in air, but do not require a handheld device can be divided into three parts: a) using wrist wearables, b) using freehand gestures, and c) using finger-wearables. Multiple wrist wearables explore text-entry on the smartwatch using the second hand [2, 23, 27, 33, 50, 79], and recognition of wrist-tilt & same-hand finger gestures [18, 21, 84]. However, none of these techniques investigate single-handed text-entry. Within freehand gestural text-entry, prior works propose hand tracking for recognizing handwritten letters or Graffiti strokes [53, 61] in air. ATK [80] demonstrates bimanual 10-finger typing in air using Leap Motion. Vulture [48] tracks a single-hand using fiducial markers and performs mid-air WGT on a distant, large screen keyboard, achieving 20.6 WPM by the 10th session (4.5hrs). However, Vulture depends on high-precision position tracking of the hand and fingers in space.

Among finger-wearables, there has been recent work on glove-typing [5, 73, 76] wherein the hand’s four fingers are instrumented with touch sensors and letters or letter groups are assigned to different finger areas which are tapped by the thumb of the same hand. The best performance was recorded by DigiTouch [73] with 16 WPM after 200mins of practice. WrisText [20] detects joystick-like wrist motion using a wide array of infrared and piezo sensors on the wrist to move a

cursor on a circular keyboard on a smartwatch. The motion is converted into text using word prediction. WristText reports speeds of 15.9 WPM after 5 days. There are other text-entry techniques that just require single-finger instrumentation using a ring [54, 65] but are applicable for typing on surfaces and not in mid-air. Other finger-worn wearables that use vision [13, 51], magnets [3, 15, 29], infrared reflectors [26, 55], motion [28, 36, 52], and touch [67, 69] do not explore text-entry. The only current single-handed ring-based typing technique in mid-air is Kim et al’s ThumbText [38], that performs a two-step letter-entry by selecting a zone and then the letter within that zone using a miniature touchpad on the finger. Kim et al. [38] additionally adapt prior Swipeboard [17] and H4 [12] techniques for the ring. In a study that simulated expert use (by investigating phrases composed of only 14 specific words), ThumbText achieved a speed of 11.4 WPM.

Typing in VR/AR

While the above techniques may potentially be useful for AR/VR, most have not been explicitly studied for it. Existing work on typing in VR includes bimanual typing using physical keyboards [39, 43, 70], augmented VR views [49] or handheld devices and controllers [22, 63]. Spiecher et al. [64] provide a good overview of VR text-entry methods. They further compare six letter-entry techniques including bimanual controller raytracing (BCR), controller tapping, freehand raytracing, a directional pad, and headpointing (raytracing from the head). All techniques use two hands except headpointing. For typing sessions of 10mins each, BCR performed best with 15 WPM, while headpointing was at 10.2 WPM. Yu et al. [81] compare letter-entry and WGT using headpointing for an HMD, and found WGT to be the fastest, reaching 24.7 WPM in the final session (80 phrases). Headpointing is a reasonable no-handed (although it requires button input for word confirmation) typing method for AR/VR. However, constant head-movement may not entirely be comfortable or preferred. Further, since the scene or the augmented objects in AR change with head movement, such typing will require the keyboard to move with the headset. This may become a constraint in scenarios where the designer wants the keyboard to be fixed within the scene.

Other approaches investigate typing for smart glasses addressing the more mobile AR contexts where controllers are not present. PalmType [72] overlays a keyboard on the palm and uses 3D tracking to locate finger taps on it. HoldBoard [1] investigates letter-entry on a smartwatch for smartglasses and reaches 10.2 WPM after 4hrs of practice. Both these letter-entry techniques require two hands. Grossman et al. [25] investigate multi-step touch-swiping on smartglasses and report an expert speed of 8.7 WPM for SwipeZone. Yu et al. [82] design 1D gestures for each letter and investigate

word-level entry where a word is inferred from a sequence of letter gestures. Upon comparison with the 1Line keyboard [42] that compresses the qwerty keyboard into a single line and supports WGT, it outperformed, reaching 9.7 WPM for expert use. Even though these are reliant on the specific smartglass affordances, the swiping techniques enable unencumbered single-handed typing for HMDs and including the ring-based ThumbText, can be considered the current benchmarks in the space that RotoSwype is tackling.

Ring Gesture Input

Besides text-entry, rings have been used to detect finger or arm movement for static and dynamic gestures. Static gestures include detection on 2D surfaces using a camera [77], in air using infrared [56], and on the ring itself using multi-touch [68]. Dynamic gestures include IMU+optical sensors for on-surface movement [37, 52]. Some works use magnets in the ring combined with wrist sensing to detect 1D twisting of the ring around the finger [4] or 2D finger tracking around a watch [30]. CyclopsRing [14] is a fish-eye camera ring that can be used to track real-time writing on the other palm. uTrack [16] performs in-air 3D pointer control using a combination of two magnetic rings on different fingers. It’s notable that none use tilt for 2D pointer control in air like RotoSwype.

3 ROTOSWYPE DESIGN

In RotoSwype, the ring’s (and therefore, the wrist’s) angular movement or rotation translates to the x-y movement of a pointer on the keyboard (Figure 2). The steps involved in typing are as follows: 1) Rotate the hand so that the pointer is on the first letter of the desired word, and click the ring’s button (Figure 2 (b)) to begin word-gesture input. 2) Rotate the hand so that the pointer traces a path over all successive letters, in order. 3) When the pointer is over the last character, click the ring’s button to end input. The top predicted word from the path traced can be seen in the input text box. 4) (Optional) To select a different word from the word suggestion box, rotate the hand to bring the pointer over the desired suggestion and click the ring’s button. The word suggestion bar area is only accessible by the pointer once the user clicks the ring’s button after performing a trace. This is crucial to preventing the pointer from overshooting the key area during wrist rotation. 5) Return to the keyboard area and start from 1) again to enter the next word. To submit the final phrase, take the pointer over the Submit key (Figure 2), and click the ring’s button. To delete a word, the user can long-press (500ms) the ring’s button. As in standard WGT, the system has no provision for per-character deletion.

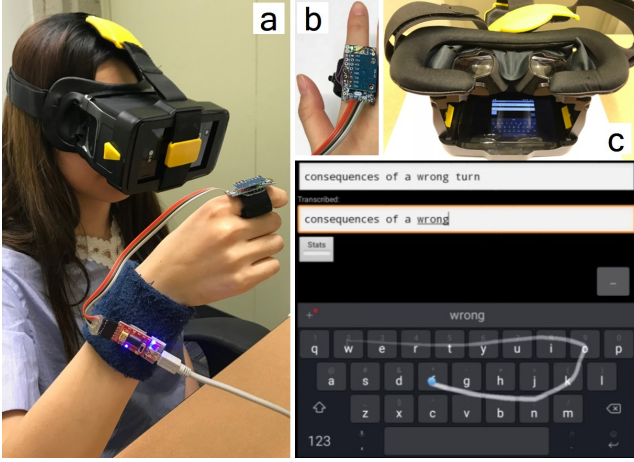


Figure 2: a) User typing using RotoSwype. b) Ring with button. c) MoGo headset with phone inside. d) The phone screen (what the user sees inside the headset). Shows the round, blue pointer drawing a trace over keyboard. The button to the right of letter *m* is designated as the Submit button.

Hardware

To build the ring (Figure 2), we used an MPU6050 [34], a 6-axis motion-tracking sensor mounted on an Arduino Pro Mini and attached to a velcro ring with a physical button. We use the MPU6050’s on-board DMP sensor fusion algorithm that processes the gyroscope and accelerometer values to provide the absolute orientation in terms of roll and pitch angles. This roll and pitch data is smoothed out using the 1-Euro filter [10] and converted into a gesture trace for the appropriate keyboard. For our exploration, we use the Nexus 4 Android smartphone encased inside the MoGo headset [45] (Figure 2c). The MoGo headset is not a VR headset, but instead contains special lenses that transfer very close objects in front of the eyes to a distance of comfort for eye convergence, thus allowing the user to see the smartphone keyboard normally at such a close distance. To utilize state-of-the-art WGT algorithms, we used the SwiftKey keyboard app and injected touch events programmatically to generate a gesture trace on the keyboard (Figure 2d). To ensure zero latency, event injection was handled via Android NDK [19] since event injection via the Android shell lags when injecting continuous touch events for the trace.

RotoSwype Design Features

Position-Control: Position control refers to mapping the range of rotational displacement to the keyboard so that the relative rotational position determines the pointer position. Velocity control refers to using the velocity of the wrist movement to control the pointer. Prior work shows the superiority of position-control against velocity-control

for tilt-based interactions [66]. Thus, similar to SWiM [78], RotoSwype employs position-control of the pointer.

Pitch and Roll: RotoSwype maps the absolute orientation of the ring along the pitch & roll dimensions to the keyboard’s x-y dimensions. Absolute orientation along pitch & roll dimensions gets calculated highly accurately since it depends on gravity. Although using yaw would open more options for Rotoswype’s design, getting absolute orientation along yaw requires a magnetometer which is susceptible to electro-magnetic interference and at times, inconsistent calibration issues.

Linear Mapping: Similar to SWiM, RotoSwype uses linear mapping to map the pitch-roll orientation to the keyboard. This is motivated by earlier work that showed linear mapping to be more accurate than quadratic for text-entry [71].

Prior work on wrist rotation control showed that users can comfortably control approximately 5° - 10° of discrete angular levels along pronation-supination and flexion-extension, and a lower resolution of control for ulnar-radial deviation [58]. While RotoSwype relies on a continuous mapping of wrist rotation to the keyboard, angular range still affects the resolution of control. We therefore based our approach on maximizing the amount of angular movement in either axis, while ensuring comfortable movement.

RotoSwype Postures

We investigate RotoSwype for two different arm postures: *hand-up* - user’s hand is above the waist and is approximately parallel to the ground, and *hand-down* - user’s hand is below the waist, orthogonal to the ground. Typing support in both postures can be useful in different scenarios. For instance, *hand-up* typing may be more conducive when sitting, and *hand-down*, when standing. For *hand-up*, we consider two wrist postures: *palm-to-the-ground* (Figure 1a centre) and *palm-to the-side* (Figure 1b centre). We now describe the postures in detail, with the ring worn on the right hand’s index finger, assuming a right-hand dominant user.

Hand-up: Palm-to-the-ground (PTG) (Figure 1a). For PTG, the wrist pronation-supination (Figure 1a left-right) motion leads to rotation along the roll and corresponds to the keyboard’s x-axis. Since pronation is more constrained than supination [24], we map an asymmetric roll angle range of -30° to 50° to the keyboard’s x-axis left-to-right. The up-down wrist bending motion (Figure 1a up-down) leads to rotation along the pitch and corresponds to the keyboard’s y-axis. The up-down wrist-bending motion is predominantly extension-flexion (face of the palm moving away or towards the hand), however at the far supination angles, it may lead to radial-ulnar deviation (side of the palm moving away or towards the hand). Again, considering the asymmetric movement

constraints [24], we map a pitch angle range of -50° to 30° to the keyboard's y-axis bottom-to-top (excluding the line with the space bar and including the suggestion bar).

Notice that both roll and pitch motion have the same 80° range, but are mapped to differing lengths on the keyboard ($x > y$). Thus the user has a lower precision of control for the pointer movement along the x-axis, compared to the y-axis. While the pronation-supination maximum range is larger [24], we cannot utilize a higher range since the hand's rotational motion for typing is across the pitch-roll axes simultaneously, and not along the individual axes separately. Thus, the user needs to be able to comfortably reach the four corners of the keyboard and all the points in between. After trying multiple ranges, we conducted a pilot with four participants. The above ranges were the maximum possible values that would allow a user to comfortably reach the four roll-pitch angles, $(-30^\circ, -50^\circ)$, $(-30^\circ, 30^\circ)$, $(50^\circ, -50^\circ)$, $(50^\circ, 30^\circ)$, corresponding to the four keyboard corners.

Hand-up: Palm-to-the-side (PTS) (Figure 1b). In PTG, the ring is worn such that the sensor is at the top of the finger. Therefore, the roll angle is 0° when the palm is to the ground. For PTS, the ring is worn to the side of the finger, and so roll angle is 0° when the palm is to the side. The pronation-supination motion still corresponds to the keyboard's x-axis, but with a symmetric and larger roll angle range of -50° to 50° (Figure 1b left-right). The up-down wrist bending motion still corresponds to the keyboard's y-axis, but is ulnar-radial deviation when the user supinates, and is extension-flexion when the user pronates. We map a smaller but symmetric pitch angle range of -35° to 35° to the keyboard's y-axis. The same process as PTG was followed to arrive at the PTS ranges.

The precision of control for the pointer movement in PTS is more consistent among the two axes, with 100° for the x-axis and 70° for y. However, PTS consists much more of the radial-ulnar deviation movement which is harder to control and perform [58]. To see which wrist posture would be better for the *hand-up* scenario, we conducted a study that will be described later.

Hand-down (Figure 1c). Typing in the hand-down posture should be much less tiring for the hand, especially if the user types in the hand-up posture without an elbow support (like a chair's handle). There is sparse research on gestures for the hand-down pose, but recent work has shown interest in the space [44, 62] owing to its relaxed nature compared to the gorilla-arm fatigue [32] associated with hand-up gestures in air.

In *hand-down*, the user's arm is orthogonal to the ground, with the palm facing the leg. Here, flexion-extension corresponds to the keyboard's x-axis with a range of -45° to 25° (Figure 1c left-right). Radial-ulnar deviation predominantly

corresponds to the y-axis with a range of -25° to 45° , however reaching the corners comfortably may require some pronation-supination as well. The range here is smaller, with 70° for both x and y, but is the maximum possible within comfort range. A second orientation where the the palm face is orthogonal to the leg was considered. However, in this case, the x-axis corresponds to radial-ulnar deviation, and it was harder to reach all four corners comfortably with any combination of mappings.

We built the RotoSwipe technique to be functional with wrist movement alone. This prevented arm fatigue that results from big hand motions. Not requiring the user to use finger motion simplifies the technique further. However, users are free to use arm and finger motion to increase the flexibility of their movements.

4 HAND-UP PRELIMINARY STUDY: PTG VS PTS

Experiment Design

We conducted a small preliminary experiment to compare PTG and PTS for their speed and accuracy. 10 right-handed participants (4F, 6M, age: $\mu = 23.9$, range=19-29) took part, 7 of whom had prior experience with word-gesture typing. The study followed a between-subjects design with 5 participants each in PTS and PTG.

Apparatus. The apparatus is the same as described above with the keyboard being 5cm wide and 2.1cm high (from just above the space bar to just above the suggestion bar since word-gesture typing introduces space automatically after word completion). The headset allows the user to wear spectacles. A long-press on the ring's hard button (500ms) invokes backspace which deletes the prior word (same as in standard word-gesture typing). To submit a phrase and go to the next one, the user takes the pointer to the Submit button and clicks the ring's button. We use the TEMA app [11] for presenting random phrases from Mackenzie et al's phrase-set [47], and logging the metrics. There was no auto-correction and next-word prediction, only the word suggestions for the path that was traced.

Procedure. Participants sat on a chair for the study duration. They were allowed to rest their elbow on the chair's arm-rest while ensuring that it did not affect typing. Each participant was first introduced to word-gesture typing and asked to practice with it for 5mins regardless of if they had used it earlier. The technique was then introduced without the headset, while the experimenter guided them. They performed one complete practice sentence with experimenter guidance following which they then wore the headset and performed one practice sentence more before starting the study. Participants were instructed to perform the task as quickly and accurately as possible. As is the norm, they were asked to

Table 1: PTG vs PTS Prelim. study result: Mean, 95% CI

Metric	PTG	PTS	ANOVA
WPM	5.5 (3.9, 7.1)	8.7 (6.1, 11.3)	$F_{1,8}=8.525, p<.05$
UER	2.7 (1.3, 4.1)	3.2 (-.9, 7.3)	n.s.
CER	19.1 (12.6, 25.6)	16.5 (12.5, 20.5)	n.s.

correct errors if they notice it immediately, but ignore them if they notice it after typing more words. They performed 10 phrases, followed by a 5min break, and then 10 phrases more. Participants were also free to relax between any two phrases and start the next phrase when they felt ready. The whole session took 45mins. Each participant performed 20 phrases, for a total of 100 phrases per technique.

Measures. We measured three metrics: *Speed*, *Uncorrected Error Rate (UER)*, *Corrected Error Rate (CER)*. Speed is measured in the standard words-per-minute (WPM) metric which starts when the user clicks the ring’s button to start the path traced and ends at the last button click before user clicks on Submit (This can be the click at the end of trace or the one to select another suggestion, whichever happens later). This includes the time spent by the user correcting errors. As in prior word-gesture typing work [85], we use UER. $UER = (MWD(S, P) \times 100) / Len(P)$, where MWD is the minimum word distance between the transcribed phrase S and target phrase P , and L is the number of words in P . Since participants performed word-level corrections by deleting a word and retyping; CER is defined as $CER = (WD \times 100) / Len(P)$, where WD is the no. of word deletions performed in the phrase. Error rates based on word distances are generally higher than string distance error rates since an error caused due to erring on a single character while gesturing will render the whole word incorrect.

PTG vs PTS: Results

A one-way anova shows that PTS’s speed is significantly greater than PTG, while UER and CER are comparable (Table 1). Even with five users, it was clear that PTS performed better than PTG. Multiple PTG participants reported that the pronation motion to left of the PTG posture was quickly fatiguing and when combined with extension, it got more uncomfortable as the phrases progressed. Based on these results, we selected PTS as the technique for the Hand-Up scenario and ran a five-day study to analyze the text-entry performance of Hand-Up (HU) PTS and Hand-Down (HD).

5 ROTOSWYPE FIVE-DAY STUDY

The goal of this study was to analyze the short-term and medium-term performance of the two RotoSwype POSTURES: HU-PTS and HD, including their learning curves. We believe that a real-world ring typing technique should be able to support both these postures. Therefore, the study is not

aimed at comparing the two techniques but at analyzing how promising both their performances are in terms of speed, accuracy, and usability.

16 right-handed participants (6F, 10M, age: $\mu = 23.5$, range=18-30), different from the prior study, took part. 12 participants had prior experience with word-gesture typing. The study followed a between-subjects design with 8 participants each in HU-PTS and HD. Each participant performed 20 phrases each day for five consecutive DAYS. Apparatus and procedure are the same as the prior study. For HU, users sat on a chair while typing. For HD, the users stood while typing with sitting during the 5min breaks. For days 2-5, participants performed one practice phrase each day after wearing the headset to ensure that they were comfortable with the setup. The overall study time over five days was 2.25hrs, with <60 mins of phrase typing. In all, 8 participants \times 5 DAYS \times 20 phrases = 800 phrases were typed for each POSTURE. We removed outlier phrases whose speed was 3 standard deviations outside the mean for a particular participant for a particular day. A total of 19 (2.3%) phrases were removed.

Results

A mixed anova was conducted for speed, UER, and CER with POSTURE as the between factor and DAY as the within factor.

Speed. Figure 3 shows the speeds over 5 days. HU-PTS starts with a speed of 8.9wpm on Day 1 to 14.4wpm on Day 5. HD starts with a speed of 9.2wpm on Day 1 to 14.8wpm on Day 5. There is no significant effect of posture on speed. However, DAY has a significant effect on speed for both HU-PTS ($F(4, 28) = 24.537, p < 0.001, \eta_p^2 = .778$) and HD ($F(4, 28) = 25.897, p < 0.001, \eta_p^2 = .787$). For HU-PTS, pairwise comparisons showed significant differences between the following day pairs: 1-3($p<.01$), 1-4($p<.005$), 1-5($p<.005$), 2-4($p<.005$), & 2-5($p<.01$). For HD: 1-2($p<.05$), 1-3($p<.05$), 1-4($p<.05$), 1-5($p<.001$), 2-5($p<.005$), & 3-5($p<.005$).

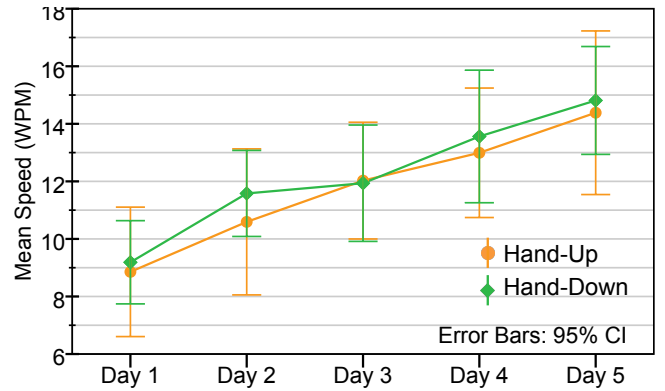


Figure 3: Mean speed (WPM) for Hand-Up (PTS) and Hand-Down over 5 days. Note that y-axis does not start at origin.

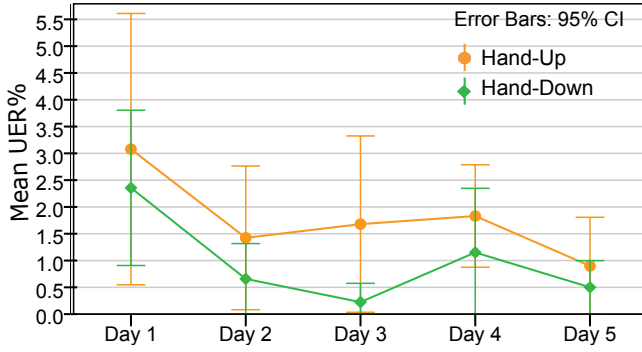


Figure 4: Mean Uncorrected Error Rate % (UER)

An almost linear growth is observed with no plateauing until the fifth day, indicating the scope for further improvement in speed. We ran four participants, two from each posture, for two additional days and their mean speeds reached 19.4 WPM (HU-PTS) and 22.2 WPM (HD) by Day 7. For perspective, regular word-gesture typists recorded word-gesture typing speeds of 34.3WPM [59] on a smartphone, 24 WPM on a smartwatch [23], and 15.5 WPM using phone-tilt for typing on the phone [78].

UER. Figure 4 shows the uncorrected error rates over 5 days. HU-PTS starts with a UER of 3.1% on Day 1 to 1.1% on Day 5. HD starts with a speed of 2.4% on Day 1 to 0.5% on Day 5. There is no significant effect of posture on UER. DAY has a significant effect on UER for HD ($F(4, 28) = 6.018, p < 0.005, \eta_p^2 = .462$), but not for HU-PTS. However, the low UER values show that in both RotoSwype postures, participants are able to rotate their wrists to follow the correct path for at least 99% of the time. Participants mentioned that some uncorrected errors were due to words that were not present in the Swiftkey dictionary, but were part of the phrase-set, such as “dewdrop”.

CER. Figure 5 shows the corrected error rates over 5 days. HU-PTS starts with a CER of 16.8% on Day 1 to 12.9% on Day 5. HD starts with a speed of 14.4% on Day 1 to 8.9% on Day 5. DAY has a significant effect on CER for both HU-PTS ($F(4, 28) = 3.515, p < .05, \eta_p^2 = .334$) and HD ($F(4, 28) = 10.001, p < .001, \eta_p^2 = .588$). These rates indicate that by Day 5, users corrected every 8th word in HU-PTS and every 11th word in HD. The low UER values compared to CER show that while participants made mistakes while tracing a word, they were able to correct that mistake and draw the correct trace upon retrying.

Subjective Ratings & Feedback. Participants rated their experience using the NASA-TLX [31] on a 7-point Likert scale on Day 1 & Day 5 (Figure 6). The load across all parameters is high on Day 1, but improves by Day 5. A Wilcoxon Signed-Rank test on HU-PTS’s Day 1 & Day 5 ratings showed that the physical demand ($Z=-2.121, p<.05$), effort ($Z=-2.049,$

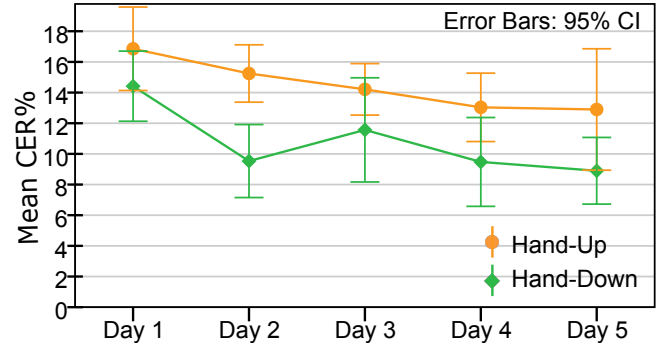


Figure 5: Mean Corrected Error Rate % (CER)

$p<.05$), and performance (as judged by the user) ($Z=-2.375, p<.05$), all significantly improved by Day 5 compared to Day 1. For HD, the mental demand ($Z=-2.328, p<.05$), physical demand ($Z=-2.342, p<.05$), effort ($Z=-2.280, p<.05$), and frustration ($Z=-2.410, p<.05$), all significantly improved by Day 5 compared to Day 1.

Participants found it difficult at first to understand the mapping of angular movements to the flat pointer motion on-screen, especially for diagonal motion. During the later days, multiple participants mentioned that they were able to draw some common words without even paying attention to the screen, hinting at muscle learning of the rotational motion - “When I see “the”, I just type it automatically. I don’t have to think or look at the keyboard”. This trend towards muscle learning indicates that the speeds may continue to improve over a long stretch as more and more words get ingrained into the muscle memory. While no explicit instructions were given, users adapted to a specific style of typing that felt comfortable to them, regardless of whether it involved finger or arm motion. While wrist-motion was the dominant movement for all, participants varied in the degree to which they used finger or arm motion.

Results Discussion

With <60mins of typing phrases, participants reached speeds of >14 WPM, with a UER of near 1% for both postures. This outperforms existing unencumbered one-handed techniques

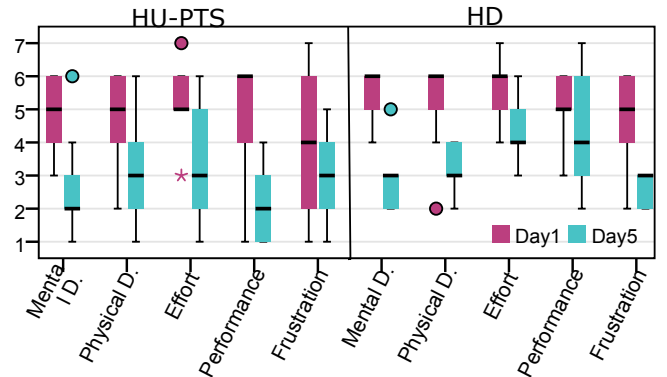


Figure 6: Subjective Ratings (Lower is better.)

Table 2: Summary: RotoSwype and the best performing technique in other modalities for HMDs. (# Both are letter-entry techniques, not word-entry. *Both use restricted word-set to simulate expert use.)

Technique	Novice WPM	Expert WPM	Novice UER%	Expert UER%
RotoSwype HU	8.9	14.4	3.1	1.1
RotoSwype HD	9.2	14.8	2.4	0.5
<i>ThumbText(Ring)</i> [#] [38]	5.5*	11.4*	13.3*	9.08*
Swipe on Glass [82]	7	9.7	0.25	0.25
<i>Controller Pointing</i> [#] [63]	15.4	NA	1	NA
Headpointing [81]	17*	24.7*	2.5*	1.9*
WrisText (non-HMD) [20]	9.9	15.2	0.2	0.1

that use ring or swipe on glass. Table 2 shows the speed and UER comparison with existing techniques focused on text-entry for HMDs that are single-handed (either unencumbered like Rotoswype, or use a hand-held device) or no-handed. The table also includes WrisText that is not investigated for HMDs, but is a relevant single-handed prediction-based technique. The novice metrics indicate the performance reported by the techniques in the first session of evaluation, while expert metrics indicate the best performance reached by the technique in the last session. Except *Controller Pointing*, all other techniques conducted multi-session evaluations with at least an hour of typing, with ThumbText and Headpointing using restricted word-sets to simulate expert performance at a faster rate. RotoSwype has the best reported novice and expert performance among unencumbered one-handed techniques that use ring or swipe on glass. Although headpointing (with word-gesture typing) uses a restricted word-set, it outperforms RotoSwype. However, as mentioned earlier, it may not be possible or preferred in all scenarios. The table only serves as a meta-comparison across different investigations much less formal than conducting a direct comparative study. However, given standard metrics of WPM and UER, we believe this is a reasonable way to contextualize the performance of RotoSwype.

6 DISCUSSION

Design Outcomes

We summarize our design outcomes here: 1) Even though yaw allows for a more direct mapping, the pitch-roll angular mapping to the keyboard plane works well and enables the technique to remain free of a magnetometer that may be prone to calibration issues and electromagnetic interference. 2) Due to limitations of the pronation+extension motion, Palm-to-Side posture is better than Palm-to-Ground for a continuous wrist tilt motion across two axes. 3) Different Linear mappings of the keyboard’s x and y-axis to the wrist rotation axes that result in different resolutions of control does not affect the eventual performance on the word-gesture

keyboard (as long as there is a minimum reasonable range. Our minimum range was 70° for the 10-key x-axis). 4) For word-gesture typing, ring-based Angular mapping is competitive with positional mapping of fingers in space. Vulture [48] which uses positional tracking using fiducial markers has a speed of 10 WPM in the first session, 15 wPM after 100 phrases, and 20.6 WPM after 400 phrases. RotoSwype has 9 WPM in the first session and 14.5 WPM after 100 phrases. Even if the RotoSwype’s performance plateaus after 16-17 WPM (which is unlikely), it is still an encouraging outcome for a self-contained ring.

Angles, Performance & Scope for Improvement

While we based our angular mapping on maximizing the angular range available to the user across the axes, it is curious that HD performs slightly better than HU-PTS, when its range is clearly smaller (70° for x-axis vs. HU-PTS’s 100°). Also, HD’s mapping is more inconsistent across x and y-axis, since 70° is mapped to both the longer x-axis and the shorter y-axis. Thus, the mapping inconsistency does not seem to affect performance (not even on Day 1). The smaller range also does not seem to have an adverse effect on HD. This may be due to a combination of two reasons: 1) Users are able to control the pointer with enough precision in the HD’s smaller range and 2) The word-gesture inference algorithm is tolerant to small path deviations. This also implies that reducing HU-PTS’s range may not harm its accuracy, and at the same time reduce the amount of required movement for the user. This warrants a dedicated investigation on the lower limits of the angular mapping to the current word-gesture keyboards. At the same time, the performance gap can also be possibly explained by the fact that the HD position is less physically fatiguing than HU.

The angular range, however, was not perfect for discrete selection on the keyboard. Multiple participants mentioned problems with clicking the submit button, for which they had to go to the precise bottom-right position and click the ring’s button. P2: “It was difficult to keep the ball (pointer) on the button, while I took my thumb to click the button.” We recognized this problem during the pilot study and to minimize it, we used the pointer position on button press, not button release. This prevented the additional deviation that the index finger would have due to the push of the thumb while clicking. However, as P2 said, the precise positioning was still not perfect. Clicking Submit took 0.9s on average, reducing WPM by 0.6 WPM. Another source of delay was the 500ms long press for deletion, which was chosen to preclude accidental deletions altogether. By day 5, users corrected every 8th word in HU-PTS and 11th word in HD (see line 628), thus the long-press reduced WPM by 0.2 and 0.15 respectively. This means that long-press and slow clicking of the Submit button reduced the WPM by about 0.8 WPM. This

is not much, but participant comments indicated that it was a prominent issue for them. We explored alternatives like double-clicking, but it interfered with quick start-stop clicks when typing words like 'a'. Multiple participants mentioned that instead of clicking the button at the start and end of the trace, a simple press-trace-release mechanism would have been better. These three issues pertaining to clicking submit, long-pressing delete, and start-stop clicking are more logistical than being a part of RotoSwype and should be fixable with a hardware designed with better resources. A simple solution to explore would be a mini-touchpad that supports different gestures for Submit & Delete. Since our results suggest that a lower wrist motion range does not necessarily reduce performance, one promising exploration would be to use a smaller range for the keyboard region and the rest for Submit and Delete options.

Wider Applicability

For our study, we explicitly changed the angular ranges when switching from HU to HD. However, it is easy to incorporate implicit detection of whether the hand is in HU or HD and change the ranges on-the-fly. While our study investigated a keyboard inside an HMD, the performance should hold for any qwerty keyboard on any device as long as the keyboard x-y ratio remains the same. For the same ratio, the mapping remains the same, regardless of the actual keyboard size. Even if the ratio deviates, a standard qwerty keyboard would probably not deviate much, and similar results should hold. Our results are based on right-handed use. While we believe that a mirrored mapping will produce similar results for left-handed use, this needs investigation.

As the use of rings gains wider acceptance, we need to formally acknowledge the unencumbered aspect of its interaction. With a ring, the hand is not holding any device and there is no device pick-up or retrieve from a pocket, but the ring is still not completely freehand. For instance, using RotoSwype while holding a coffee or a heavy bag is probably a bad idea, but many small everyday items such as keys, coins, transit tokens, and credit cards, can be held comfortably in the three remaining fingers while performing this style of text-entry.

Beyond this, future work needs to look into the following areas: combining RotoSwype with a letter-entry technique so as to enable typing out-of-vocabulary (OOV) words; including the provision for number and symbol entry; and exploring RotoSwype or similar techniques for when the user is walking.

7 CONCLUSION

In this paper we introduced RotoSwype, a ring-tilt based text-entry technique that enables unencumbered, one-handed,

self-contained, and eyes-away typing that enables word-gesture typing on any qwerty keyboard on any device. We design and build the technique for the Hand-Up and Hand-Down postures. For Hand-Up, we further compare two base-poses of the hand, PTS and PTG, and choose PTS based on a preliminary study. In a 5-day study, we evaluate RotoSwype for text-entry for AR/VR HMDs. The results show that with <60 mins of typing, participants achieve speeds of >14 WPM with a near 1% uncorrected error rate for both HU and HD postures, outperforming existing unencumbered techniques that use ring-input or swiping on glasses. We further discuss subjective feedback, design outcomes, improvements, and wider applicability for future work.

8 ACKNOWLEDGEMENTS

This work made possible by NSERC Discovery Grant #2018-05187, the Canada Foundation for Innovation Infrastructure Fund "Facility for Fully Interactive Physio-digital Spaces" (#33151), and Ontario Early Researcher Award #ER16-12-184.

REFERENCES

- [1] Sunggeun Ahn, Seongkook Heo, and Geehyuk Lee. 2017. Typing on a Smartwatch for Smart Glasses. In *Proc. Interact. Surfaces Spaces - ISS '17*. ACM Press, New York, New York, USA, 201–209. <https://doi.org/10.1145/3132272.3134136>
- [2] Ahmed Sabbir Arif and Ali Mazalek. 2016. A survey of text entry techniques for smartwatches. In *International Conference on Human-Computer Interaction*. Springer, 255–267.
- [3] Daniel Ashbrook, Patrick Baudisch, and Sean White. 2011. Nanya: subtle and eyes-free mobile input with a magnetically-tracked finger ring. In *Proc. 2011 Annu. Conf. Hum. factors Comput. Syst. - CHI '11*. ACM Press, New York, New York, USA, 2043. <https://doi.org/10.1145/1978942.1979238>
- [4] Daniel Ashbrook, Patrick Baudisch, and Sean White. 2011. Nanya: Subtle and Eyes-free Mobile Input with a Magnetically-tracked Finger Ring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2043–2046. <https://doi.org/10.1145/1978942.1979238>
- [5] Bartosz Bajer, I Scott MacKenzie, and Melanie Baljko. 2012. Huffman base-4 text entry glove (H4 TEG). In *2012 16th International Symposium on Wearable Computers*. IEEE, 41–47.
- [6] Nikola Banovic, Koji Yatani, and Khai N Truong. 2013. Escape-keyboard: A sight-free one-handed text entry method for mobile touch-screen devices. *International Journal of Mobile Human Computer Interaction (IJMHCI)* 5, 3 (2013), 42–61.
- [7] Thomas Baudel and Michel Beaudouin-Lafon. 1993. Charade: remote control of objects using free-hand gestures. *Commun. ACM* 36, 7 (jul 1993), 28–35. <https://doi.org/10.1145/159544.159562>
- [8] Xiaojun Bi and Shumin Zhai. 2016. IJQwerty: What Difference Does One Key Change Make? Gesture Typing Keyboard Optimization Bounded by One Key Position Change from Qwerty. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 49–58. <https://doi.org/10.1145/2858036.2858421>
- [9] Matthew N. Bonner, Jeremy T. Brudvik, Gregory D. Abowd, and W. Keith Edwards. 2010. No-Look Notes: Accessible Eyes-Free Multi-touch Text Entry. Springer, Berlin, Heidelberg, 409–426. https://doi.org/10.1007/978-3-642-12654-3_24

- [10] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1 & #8364; Filter: A Simple Speed-based Low-pass Filter for Noisy Input in Interactive Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2527–2530. <https://doi.org/10.1145/2207676.2208639>
- [11] Steven J Castellucci and I Scott MacKenzie. 2011. Gathering text entry metrics on android devices. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. ACM, 1507–1512.
- [12] Steven J Castellucci and I Scott MacKenzie. 2013. Gestural text entry using Huffman codes. In *Proceedings of the International Conference on Multimedia and Human-Computer Interaction*, Vol. 119. 1–8.
- [13] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. 2015. CyclopsRing: Enabling Whole-Hand and Context-Aware Interactions Through a Fisheye Ring. In *Proc. 28th Annu. ACM Symp. User Interface Softw. Technol. - UIST '15*. ACM Press, New York, New York, USA, 549–556. <https://doi.org/10.1145/2807442.2807450>
- [14] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. 2015. CyclopsRing: Enabling Whole-Hand and Context-Aware Interactions Through a Fisheye Ring. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & #38; Technology (UIST '15)*. ACM, New York, NY, USA, 549–556. <https://doi.org/10.1145/2807442.2807450>
- [15] Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. 2013. uTrack: 3D input using two magnetic sensors. In *Proc. 26th Annu. ACM Symp. User interface Softw. Technol. - UIST '13*. ACM Press, New York, New York, USA, 237–244. <https://doi.org/10.1145/2501988.2502035>
- [16] Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. 2013. uTrack: 3D Input Using Two Magnetic Sensors. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 237–244. <https://doi.org/10.1145/2501988.2502035>
- [17] Xiang 'Anthony' Chen, Tovi Grossman, and George Fitzmaurice. 2014. Swipeboard. In *Proc. 27th Annu. ACM Symp. User interface Softw. Technol. - UIST '14*. ACM Press, New York, New York, USA, 615–620. <https://doi.org/10.1145/2642918.2647354>
- [18] Artem Dementyev and Joseph A. Paradiso. 2014. WristFlex. In *Proc. 27th Annu. ACM Symp. User interface Softw. Technol. - UIST '14*. ACM Press, New York, New York, USA, 161–166. <https://doi.org/10.1145/2642918.2647396>
- [19] Android Developers. 2018. Android NDK. (2018). <https://developer.android.com/ndk/>
- [20] Jun Gong, Zheer Xu, Qifan Guo, Teddy Seyed, Xiang 'Anthony' Chen, Xiaojun Bi, and Xing-Dong Yang. 2018. WrisText: One-handed Text Entry on Smartwatch Using Wrist Gestures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 181, 14 pages. <https://doi.org/10.1145/3173574.3173755>
- [21] Jun Gong, Xing-Dong Yang, and Pourang Irani. 2016. WristWhirl: One-handed Continuous Smartwatch Input using Wrist Gestures. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 861–872.
- [22] Gabriel González, José P Molina, Arturo S García, Diego Martínez, and Pascual González. 2009. Evaluation of text input techniques in immersive virtual environments. In *New Trends on Human-Computer Interaction*. Springer, 109–118.
- [23] Mitchell Gordon, Tom Ouyang, and Shumin Zhai. 2016. WatchWriter: Tap and Gesture Typing on a Smartwatch Miniature Keyboard with Statistical Decoding. In *Proc. 2016 CHI Conf. Hum. Factors Comput. Syst. - CHI '16*. ACM Press, New York, New York, USA, 3817–3821. <https://doi.org/10.1145/2858036.2858242>
- [24] E. (Etienne) Grandjean and Etienne. 1988. *Fitting the task to the man : a textbook of occupational ergonomics*. Taylor & Francis. 363 pages. <https://dl.acm.org/citation.cfm?id=76694>
- [25] Tovi Grossman, Xiang Anthony Chen, and George Fitzmaurice. 2015. Typing on Glasses: Adapting Text Entry to Smart Eyewear. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '15)*. ACM, New York, NY, USA, 144–152. <https://doi.org/10.1145/2785830.2785867>
- [26] Aakar Gupta, Muhammed Anwar, and Ravin Balakrishnan. 2016. Porous Interfaces for Small Screen Multitasking using Finger Identification. In *Proc. 29th Annu. Symp. User Interface Softw. Technol. - UIST '16*. ACM Press, New York, New York, USA, 145–156. <https://doi.org/10.1145/2984511.2984557>
- [27] Aakar Gupta and Ravin Balakrishnan. 2016. DualKey: miniature screen text entry via finger identification. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 59–70.
- [28] Aakar Gupta, Jiushan Yang, and Ravin Balakrishnan. 2018. Asterisk and Obelisk: Motion Codes for Passive Tagging. In *Proc. 25th Annu. ACM Symp. User interface Softw. Technol. - UIST '18*. ACM.
- [29] Chris Harrison and Scott E. Hudson. 2009. Abracadabra. In *Proc. 22nd Annu. ACM Symp. User interface Softw. Technol. - UIST '09*. ACM Press, New York, New York, USA, 121. <https://doi.org/10.1145/1622176.1622199>
- [30] Chris Harrison and Scott E. Hudson. 2009. Abracadabra: Wireless, High-precision, and Unpowered Finger Input for Very Small Mobile Devices. In *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology (UIST '09)*. ACM, New York, NY, USA, 121–124. <https://doi.org/10.1145/1622176.1622199>
- [31] Sandra G Hart. 1986. NASA Task load Index (TLX). Volume 1.0; Paper and pencil package. (1986).
- [32] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-air Interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1063–1072. <https://doi.org/10.1145/2556288.2557130>
- [33] Jonggi Hong, Seongkook Heo, Poika Isokoski, and Geehyuk Lee. 2015. SplitBoard: A simple split soft keyboard for wristwatch-sized touch screens. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1233–1236.
- [34] Invensense. 2018. MPU-6050 Motion Tracking | TDK. (2018). <https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>
- [35] Eleanor Jones, Jason Alexander, Andreas Andreou, Pourang Irani, and Sriram Subramanian. 2010. GesText: accelerometer-based gestural text-entry systems. In *Proc. 28th Int. Conf. Hum. factors Comput. Syst. - CHI '10*. ACM Press, New York, New York, USA, 2173. <https://doi.org/10.1145/1753326.1753655>
- [36] Wolf Kienzle and Ken Hinckley. 2014. LightRing: always-available 2D input on any surface. In *Proc. 27th Annu. ACM Symp. User interface Softw. Technol. - UIST '14*. ACM Press, New York, New York, USA, 157–160. <https://doi.org/10.1145/2642918.2647376>
- [37] Wolf Kienzle and Ken Hinckley. 2014. LightRing: Always-available 2D Input on Any Surface. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 157–160. <https://doi.org/10.1145/2642918.2647376>
- [38] Junhyeok Kim, William Delamare, and Pourang Irani. 2018. ThumbText: Text Entry for Wearable Devices Using a Miniature Ring. In *Proceedings of Graphics Interface 2018 (GI 2018)*. Canadian Human-Computer Communications Society / Société canadienne du dialogue humain-machine, 18 – 25. <https://doi.org/10.20380/GI2018.04>
- [39] Pascal Knierim, Valentin Schwind, Anna Maria Feit, Florian Nieuwenhuizen, and Niels Henze. 2018. Physical Keyboards in Virtual Reality. In *Proc. 2018 CHI Conf. Hum. Factors Comput. Syst. - CHI '18*. ACM Press, New York, New York, USA, 1–9. <https://doi.org/10.1145/3173574.3173919>

- [40] Per Ola Kristensson. 2007. *Discrete and continuous shape writing for text entry and control*. Ph.D. Dissertation.
- [41] Per-Ola Kristensson and Shumin Zhai. 2004. SHARK2: A Large Vocabulary Shorthand Writing System for Pen-based Computers. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, New York, NY, USA, 43–52. <https://doi.org/10.1145/1029632.1029640>
- [42] Frank Chun Yat Li, Richard T. Guy, Koji Yatani, and Khai N. Truong. 2011. The 1Line Keyboard: A QWERTY Layout in a Single Line. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 461–470. <https://doi.org/10.1145/2047196.2047257>
- [43] Jia-Wei Lin, Ping-Hsuan Han, Jiun-Yu Lee, Yang-Sheng Chen, Ting-Wei Chang, Kuan-Wen Chen, and Yi-Ping Hung. 2017. Visualizing the Keyboard in Virtual Reality for Enhancing Immersive Experience. In *ACM SIGGRAPH 2017 Posters (SIGGRAPH '17)*. ACM, New York, NY, USA, Article 35, 2 pages. <https://doi.org/10.1145/3102163.3102175>
- [44] Mingyu Liu, Mathieu Nancel, and Daniel Vogel. 2015. Gunslinger: Subtle Arms-down Mid-air Interaction. In *Proc. 28th Annu. ACM Symp. User Interface Softw. Technol. - UIST '15*. ACM Press, New York, New York, USA, 63–71. <https://doi.org/10.1145/2807442.2807489>
- [45] Cinema2Go LTD. 2018. MOGO Travel. (2018). <https://www.cinema2go.biz/mogo-travel>
- [46] Kent Lyons, Thad Starner, Daniel Plaisted, James Fusia, Amanda Lyons, Aaron Drew, and E. W. Looney. 2004. Twiddler typing. In *Proc. 2004 Conf. Hum. factors Comput. Syst. - CHI '04*. ACM Press, New York, New York, USA, 671–678. <https://doi.org/10.1145/985692.985777>
- [47] I Scott MacKenzie and R William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *CHI'03 extended abstracts on Human factors in computing systems*. ACM, 754–755.
- [48] Anders Markussen, Mikkel Rønne Jakobsen, Kasper Hornbæk, Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. 2014. Vulture: a mid-air word-gesture keyboard. In *Proc. 32nd Annu. ACM Conf. Hum. factors Comput. Syst. - CHI '14*. ACM Press, New York, New York, USA, 1073–1082. <https://doi.org/10.1145/2556288.2556964>
- [49] Mark McGill, Daniel Boland, Roderick Murray-Smith, and Stephen Brewster. 2015. A Dose of Reality: Overcoming Usability Challenges in VR Head-Mounted Displays. In *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst. - CHI '15*. ACM Press, New York, New York, USA, 2143–2152. <https://doi.org/10.1145/2702123.2702382>
- [50] Aske Mottelson, Christoffer Larsen, Mikkel Lyderik, Paul Strohmeier, and Jarrod Knibbe. 2016. Invisiboard: maximizing display and input space with a full screen text entry method for smartwatches. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 53–59.
- [51] Suranga Nanayakkara, Roy Shilkrot, and Pattie Maes. 2012. EyeRing: an eye on a finger. In *Proc. 2012 ACM Annu. Conf. Ext. Abstr. Hum. Factors Comput. Syst. Ext. Abstr. - CHI EA '12*. ACM Press, New York, New York, USA, 1961. <https://doi.org/10.1145/2212776.2223736>
- [52] A. Nguyen and A. Banic. 2015. 3D Touch: A wearable 3D input device for 3D applications. In *2015 IEEE Virtual Reality (VR)*. 55–61. <https://doi.org/10.1109/VR.2015.7223324>
- [53] Tao Ni, Doug Bowman, and Chris North. 2011. AirStroke: bringing unistroke text entry to freehand gesture interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2473–2476.
- [54] Shahriar Nirjon, Jeremy Gummeson, Dan Gelb, and Kyu-Han Kim. 2015. TypingRing: A Wearable Ring Platform for Text Input. In *Proc. 13th Annu. Int. Conf. Mob. Syst. Appl. Serv. - MobiSys '15*. ACM Press, New York, New York, USA, 227–239. <https://doi.org/10.1145/2742647.2742665>
- [55] Masa Ogata, Yuta Sugiura, Hirotaka Osawa, and Michita Imai. 2012. iRing: intelligent ring using infrared reflection. In *Proc. 25th Annu. ACM Symp. User interface Softw. Technol. - UIST '12*. ACM Press, New York, New York, USA, 131. <https://doi.org/10.1145/2380116.2380135>
- [56] Masa Ogata, Yuta Sugiura, Hirotaka Osawa, and Michita Imai. 2012. iRing: Intelligent Ring Using Infrared Reflection. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 131–136. <https://doi.org/10.1145/2380116.2380135>
- [57] Kurt Partridge, Saurav Chatterjee, Vibha Sazawal, Gaetano Borriello, and Roy Want. 2002. TiltType: accelerometer-supported text entry for very small devices. In *Proc. 15th Annu. ACM Symp. User interface Softw. Technol. - UIST '02*. ACM Press, New York, New York, USA, 201. <https://doi.org/10.1145/571985.572013>
- [58] Mahfuz Rahman, Sean Gustafson, Pourang Irani, and Sriram Subramanian. 2009. Tilt techniques: Investigating the Dexterity of Wrist-based Input. In *Proc. 27th Int. Conf. Hum. factors Comput. Syst. - CHI '09*. ACM Press, New York, New York, USA, 1943. <https://doi.org/10.1145/1518701.1518997>
- [59] Shyam Rey, Shumin Zhai, and Per Ola Kristensson. 2015. Performance and User Experience of Touchscreen and Gesture Keyboards in a Lab Setting and in the Wild. In *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst. - CHI '15*. ACM Press, New York, New York, USA, 679–688. <https://doi.org/10.1145/2702123.2702597>
- [60] Vibha Sazawal, Roy Want, and Gaetano Borriello. 2002. The unigesture approach one-handed text entry for small devices. In *International Conference on Mobile Human-Computer Interaction*. Springer, 256–270.
- [61] Alexander Schick, Daniel Morlock, Christoph Amma, Tanja Schultz, and Rainer Stiefelhausen. 2012. Vision-based handwriting recognition for unrestricted text input in mid-air. In *Proceedings of the 14th ACM international conference on Multimodal interaction*. ACM, 217–220.
- [62] Shaishav Siddhpuria, Keiko Katsuragawa, James R. Wallace, and Edward Lank. 2017. Exploring At-Your-Side Gestural Interaction for Ubiquitous Environments. In *Proceedings of the 2017 Conference on Designing Interactive Systems (DIS '17)*. ACM, New York, NY, USA, 1111–1122. <https://doi.org/10.1145/3064663.3064695>
- [63] Marco Speicher, Anna Maria Feit, Pascal Ziegler, and Antonio Krüger. 2018. Selection-based Text Entry in Virtual Reality. In *Proc. 2018 CHI Conf. Hum. Factors Comput. Syst. - CHI '18*. ACM Press, New York, New York, USA, 1–13. <https://doi.org/10.1145/3173574.3174221>
- [64] Marco Speicher, Anna Maria Feit, Pascal Ziegler, and Antonio Krüger. 2018. Selection-based Text Entry in Virtual Reality. In *Proc. 2018 CHI Conf. Hum. Factors Comput. Syst. - CHI '18*. ACM Press, New York, New York, USA, 1–13. <https://doi.org/10.1145/3173574.3174221>
- [65] Tap Systems. 2018. Tap - Wearable Keyboard, Mouse & Controller. (2018). <https://www.tapwithus.com/>
- [66] Robert J Teather and I Scott MacKenzie. 2014. Position vs. velocity control for tilt-based interaction. In *Proceedings of Graphics Interface 2014*. Canadian Information Processing Society, 51–58.
- [67] Hsin-Ruey Tsai, Min-Chieh Hsiu, Jui-Chun Hsiao, Lee-Ting Huang, Mike Chen, and Yi-Ping Hung. 2016. TouchRing: subtle and always-available input using a multi-touch ring. In *Proc. 18th Int. Conf. Human-Computer Interact. with Mob. Devices Serv. Adjunct - MobileHCI '16*. ACM Press, New York, New York, USA, 891–898. <https://doi.org/10.1145/2957265.2961860>
- [68] Hsin-Ruey Tsai, Min-Chieh Hsiu, Jui-Chun Hsiao, Lee-Ting Huang, Mike Chen, and Yi-Ping Hung. 2016. TouchRing: Subtle and Always-available Input Using a Multi-touch Ring. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '16)*. ACM, New York, NY, USA, 891–898. <https://doi.org/10.1145/2957265.2961860>

- [69] Koji Tsukada and Michiaki Yasumura. 2001. Ubi-finger: Gesture input device for mobile use. In *UbiComp 2001 Informal Companion Proceedings*. 11.
- [70] James Walker, Bochao Li, Keith Vertanen, and Scott Kuhl. 2017. Efficient Typing on a Visually Occluded Physical Keyboard. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 5457–5461. <https://doi.org/10.1145/3025453.3025783>
- [71] William S. Walmsley, W. Xavier Snelgrove, and Khai N. Truong. 2014. Disambiguation of imprecise input with one-dimensional rotational text entry. *ACM Trans. Comput. Interact.* 21, 1 (feb 2014), 1–40. <https://doi.org/10.1145/2542544>
- [72] Guan Wang and Ayoung Suh. 2018. Disorder or Driver?. In *Proc. 2018 CHI Conf. Hum. Factors Comput. Syst. - CHI '18*. ACM Press, New York, New York, USA, 1–12. <https://doi.org/10.1145/3173574.3173624>
- [73] Eric Whitmire, Mohit Jain, Divye Jain, Greg Nelson, Ravi Karkar, Shwetak Patel, and Mayank Goel. 2017. DigiTouch. *Proc. ACM Interactive, Mobile, Wearable Ubiquitous Technol.* 1, 3 (sep 2017), 1–21. <https://doi.org/10.1145/3130978>
- [74] Daniel Wigdor and Ravin Balakrishnan. 2003. TiltText: using tilt for text input to mobile phones. In *Proc. 16th Annu. ACM Symp. User interface Softw. Technol. - UIST '03*. ACM Press, New York, New York, USA, 81–90. <https://doi.org/10.1145/964696.964705>
- [75] John Williamson and Roderick Murray-Smith. 2005. Hex: Dynamics and probabilistic text entry. In *Switching and Learning in Feedback Systems*. Springer, 333–342.
- [76] Pui Chung Wong, Kening Zhu, and Hongbo Fu. 2018. FingerT9: Leveraging Thumb-to-finger Interaction for Same-side-hand Text Entry on Smartwatches. In *Proc. 2018 CHI Conf. Hum. Factors Comput. Syst. - CHI '18*. ACM Press, New York, New York, USA, 1–10. <https://doi.org/10.1145/3173574.3173752>
- [77] Xing-Dong Yang, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. 2012. Magic Finger: Always-available Input Through Finger Instrumentation. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 147–156. <https://doi.org/10.1145/2380116.2380137>
- [78] Hui-Shyong Yeo, Xiao-Shen Phang, Steven J. Castellucci, Per Ola Kristensson, and Aaron Quigley. 2017. Investigating Tilt-based Gesture Keyboard Entry for Single-Handed Text Entry on Large Devices. In *Proc. 2017 CHI Conf. Hum. Factors Comput. Syst. - CHI '17*. ACM Press, New York, New York, USA, 4194–4202. <https://doi.org/10.1145/3025453.3025520>
- [79] Xin Yi, Chun Yu, Weijie Xu, Xiaojun Bi, and Yuanchun Shi. 2017. COMPASS: Rotational Keyboard on Non-Touch Smartwatches. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 705–715.
- [80] Xin Yi, Chun Yu, Mingrui Zhang, Sida Gao, Ke Sun, and Yuanchun Shi. 2015. Atk: Enabling ten-finger freehand typing in air based on 3d hand tracking data. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 539–548.
- [81] Chun Yu, Yizheng Gu, Zhican Yang, Xin Yi, Hengliang Luo, and Yuanchun Shi. 2017. Tap, Dwell or Gesture?: Exploring Head-Based Text Entry Techniques for HMDs. In *Proc. 2017 CHI Conf. Hum. Factors Comput. Syst. - CHI '17*. ACM Press, New York, New York, USA, 4479–4488. <https://doi.org/10.1145/3025453.3025964>
- [82] Chun Yu, Ke Sun, Mingyuan Zhong, Xincheng Li, Peijun Zhao, and Yuanchun Shi. 2016. One-Dimensional Handwriting. In *Proc. 2016 CHI Conf. Hum. Factors Comput. Syst. - CHI '16*. ACM Press, New York, New York, USA, 71–82. <https://doi.org/10.1145/2858036.2858542>
- [83] Shumin Zhai and Per-Ola Kristensson. 2003. Shorthand Writing on Stylus Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 97–104. <https://doi.org/10.1145/642611.642630>
- [84] Yang Zhang and Chris Harrison. 2015. Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 167–173.
- [85] Suwen Zhu, Tianyao Luo, Xiaojun Bi, and Shumin Zhai. 2018. Typing on an Invisible Keyboard. In *Proc. 2018 CHI Conf. Hum. Factors Comput. Syst. - CHI '18*. ACM Press, New York, New York, USA, 1–13. <https://doi.org/10.1145/3173574.3174013>